REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
	Technical Report	-

4. TITLE AND SUBTITLE
Sparsity and Nullity: Paradigm for Analysis Dictionary Learning

5a. CONTRACT NUMBER

5b. GRANT NUMBER
W911NF-04-D-0003

5c. PROGRAM ELEMENT NUMBER
611102

6. AUTHORS
Xiao Bian, Hamid Krim, Alex Bronstein, Liyi Dai

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
North Carolina State University
2701 Sullivan Drive
Admin Services III; Box 7514
Raleigh, NC 27695 -7514

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
62483-CS-SR.14

12. DISTRIBUTION AVAILIBILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT
Sparse models in dictionary learning have been successfully applied in a wide variety of machine learning and computer vision problems, and have also recently emerged with increasing research interest. Another interesting related problem based on linear equality constraint, namely the sparse null space problem (SNS), first appeared in 1986 and has since inspired results on sparse basis pursuit.
In this paper, we investigate the relation between the SNS problem and the analysis dictionary learning problem, and show that the SNS problem plays a central role, and may be utilized to solve dictionary learning problems

15. SUBJECT TERMS
Dictionary learning, sparse coding, sparse null space problem

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Liyi Dai
UU	UU	UU	UU		19b. TELEPHONE NUMBER 919-549-4350

Report Title

Sparsity and Nullity: Paradigm for Analysis Dictionary Learning

ABSTRACT

Sparse models in dictionary learning have been successfully applied in a wide variety of machine learning and computer vision problems, and have also recently emerged with increasing research interest. Another interesting related problem based on linear equality constraint, namely the sparse null space problem (SNS), first appeared in 1986 and has since inspired results on sparse basis pursuit.

In this paper, we investigate the relation between the SNS problem and the analysis dictionary learning problem, and show that the SNS problem plays a central role, and may be utilized to solve dictionary learning problems. Moreover, we propose an efficient algorithm of sparse null space basis pursuit, and extend it to a solution of analysis dictionary learning. Experimental results on numerical synthetic data and real-world data are further presented to validate the performance of our method.

SPARSITY AND NULLITY: PARAGIDMS FOR ANALYSIS DICTIONARY LEARNING

XIAO BIAN*, HAMID KRIM†, ALEX BRONSTEIN‡, AND LIYI DAI§

Abstract. Sparse models in dictionary learning have been successfully applied in a wide variety of machine learning and computer vision problems, and have also recently emerged with increasing research interest. Another interesting related problem based on linear equality constraint, namely the sparse null space problem (SNS), first appeared in 1986 and has since inspired results on sparse basis pursuit.

In this paper, we investigate the relation between the SNS problem and the analysis dictionary learning problem, and show that the SNS problem plays a central role, and may be utilized to solve dictionary learning problems. Moreover, we propose an efficient algorithm of sparse null space basis pursuit, and extend it to a solution of analysis dictionary learning. Experimental results on numerical synthetic data and real-world data are further presented to validate the performance of our method.

Key words. Dictionary learning, sparse coding, sparse null space problem

AMS subject classifications. 65F10, 90C25

1. Introduction.

High dimensional data analysis has been the focus of research in diverse areas, including machine learning, computer vision, and applied mathematics, on account of its theoretical complexity, and its high relevance to big data problems. Dictionary learning has been one of the key methodologies in addressing high dimensional data, and has successfully been applied in feature extraction [12], for signal denoising [12] [10] [24], pattern recognition and classification [20], etc.

Specifically, a set of atoms learned from a given dataset are considered as a dictionary, and are expected to have the potential to analyze unknown incoming data. In order to construct an effective dictionary, signal models play a key role. One common assumption is that high dimensional data is concentrated in a low-dimensional manifold embedded in a high dimensional space. Dimension reduction was hence a natural way to characterize the data, and was subsequently extended to a larger family of algorithms, and often referred to as nonlinear dimension reduction [23][9]. With a different perspective on the same issue, recent research has shown that sparse models may also be very useful for learning discriminating and robust dictionaries from data (see [10] and references within). While these were broadly applicable, a particularly well adapted structure of data, so-called a union of subspaces (UoS), was revealing of the power of such approaches. In particular, using a parsimony constraint on the number of atoms to represent the data at hand, helps effectively recover the underlying basis of each subspace [13] [3]. In dictionary learning, *synthesis* and *analysis* models were proposed, and their respective strengths and limitations in obtaing sparse representations of data, are also of interest. On the one hand, synthesis models seek a synthesis dictionary $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_n]$ such that $\mathbf{x}_i = \sum_{j \in S} \mathbf{d}_j w_{ij}, \|S\|_0 \leq k$, where $\mathbf{x}_i$ is the data of interest, $\mathbf{d}_j$ is the j^{th} atom in the dictionary, and w_{ij} the associated coefficient. On the other hand, an operator H

*Department of Electrical Engineering, North Carolina State University, Raleigh, North Carolina, USA, (xbian@ncsu.edu).

†Department of Electrical Engineering, North Carolina State University, Raleigh, North Carolina, USA, (ahk@ncsu.edu).

‡School of Electrical Engineering, Tel Aviv University, Tel Aviv, Israel, (bron@eng.tau.ac.il)

§Computing Sciences Division, U.S. Army Research Office, Research Triangle Park, NC, USA, (liyi.dai.civ@mail.mil)

is sought in an analysis model, so that $H \circ \mathbf{x}_i$ yields a sparse coefficient vector for representing $\mathbf{x}_i$ [2][24].

Another interesting and seemingly unrelated problem, invoking sparsity is the sparse null space problem (SNS), and was first proposed in 1986 by Coleman and Pothen [8]. As we elaborate further in this paper, the SNS problem may be stated as searching for a sparse basis for the null space of a given matrix $\mathbf{A}$. We demonstrate that the sparse null space solution is instrumental in helping understand the analysis dictionary learning problem, and in providing sufficient insight for achieving systematic and applicable solutions.

In this paper, we examine the relation between the SNS problem and the dictionary learning problem, and we prove that the SNS problem is equivalent to the analysis dictionary learning problem (ADL). We then proceed to solve the ADL problem using methods for the SNS problem. Specifically, inspired by the existing results for SNS problem and the state-of-the-art of sparsity pursuit algorithms, we present a l_1 minimization-based greedy algorithm to solve the SNS problem. In contrast to current mainstream algorithms [10][24][20][15], the convergence of our method is assured by both the convergence of the greedy algorithm and the convex l_1 minimization. Moreover, we demonstrate its superior performance on both synthetic numerical data and real-world data.

The remainder of this paper is organized as follows. In Section 2, we discuss the current state of the art in the area of dictionary learning, specifically in ADL. In Section 3, we analyze the relationship of the SNS problem to that of ADL, and show their equivalence. In Section 4, we present an effective method to solve the SNS problem, and which essentially can also efficiently solve the ADL problem. Finally, in Section 5 we validate our method on the analysis dictionary learning problem by numerical experiments, and illustrate the effectiveness of our algorithm on texture classification.

1.1. Notation. The notational conventions used throughout this paper are as follows: For an $m \times n$ matrix $\mathbf{X}$, the space spanned by its rows is denoted as row($\mathbf{X}$), and that spanned by its columns col($\mathbf{X}$). Its null space is denoted by null($\mathbf{X}$). The sparsity of $\mathbf{X}$, defined as $\frac{\|\mathbf{X}\|_0}{mn}$, is denoted by $\rho(\mathbf{X})$. Moreover, we denote by $P_{\mathbf{X}}$ the projection matrix onto col($\mathbf{X}$), and by $P_{\mathbf{X}^\perp} = \mathbf{I} - P_{\mathbf{X}}^T P_{\mathbf{X}}$ the projection matrix onto null($\mathbf{X}$). Additionally, given a vector $\mathbf{y} \in R^n$, operator $(\cdot)_j$ returns the value of the j^{th} element of $\mathbf{y}$. The adjoint operator of $(\cdot)_j$, denoted as $(\cdot)_j^*$, is hence as follows: $(c)_j^* = \mathbf{v} \in R^n$, such that $(\mathbf{v})_j = c$ and $(\mathbf{v})_i = 0$, if $i \neq j$. Finally, we denote the set of all sparse vectors $\{\mathbf{x} | \|\mathbf{x}\|_0 \leq k\}$ by Σ_k.

2. Related work. Much of the reseach in dictionary learning has primarily focused on synthesis models such as [1][20][19][15], among many others. Remarkable performance has been achieved in learning effective dictionaries which are well adapted to specific datasets, especially on imagery data. The contribution of the atoms in the representation of the the training data was constrained to be sparse. The same sparseness is exploited to recover an input signal from corrupted data [1] or to classify signals into different clusters [14].

Specifically, the learned dictionary and the corresponding sparse coefficients are alternately updated untill convergence or until a target performance attainment. With the sparse coefficients in hand, learning each atom may be solved by the gradient descent or its variations, such as the stochastic gradient descent [4]. With the dictionary atoms discovery, different approaches have been proposed to determine the sparse coefficients. In K-SVD [1], for example, orthogonal matching pursuit(OMP) is used to recover the sparse coefficient, while Lasso-LARS is chosen in online dictionary learning for sparse coding [20].

A well known issue with synthesis dictionary learning is the poor stability in its signal repre-

sentation [25][11]. This is primarily due to the difficulty in controlling the coherence of the learned dictionary, which may, in turn, lead to multiple representations of the same signal [5]. While this phenomenon does not particularly adversely affect denoising [11], it may significantly impact a consistent classification or clustering performance. In addition, the computational cost to process new incoming data points, often calls for a procedure such as a matching pursuit procedure [1], or a sparse coding routine such as Lasso [20], is too significant for purposed of large high dimensional data sets.

While ADL is known to be difficult to train, it is free of the above-noted SDL limitations [25]. Additionally, upon learning the dictionary $\mathbf{D}$, the representation of any data $\mathbf{x}$ is unique as it is the result of one matrix-vector multiplication. The latter fact also yields a low computational cost of processing new data which is both linear in the data dimension and in the sample size.

Analysis K-SVD [25] provides, to the best of our knowledge, the current state-of-the-art solution to the ADL problem. In the learning procedure, a framework similar to K-SVD is designed to alternately update the dictionary and the sparse coefficients. At each iteration, each atom is independently updated by minimizing the covariance between the atom and a subset of data samples that are "almost orthogonal" to the atom. The minimization may be formulated as searching for the singular vectors corresponding to the minimal singular value of the data matrix of the subset of samples. Analysis K-SVD generally achieves the best known performance in the recovery of the original data space.

As a departure from Analysis K-SVD, we present in this paper an alternative ADL algorithm which is closely related to SNS problem discussed next. By decomposing the DL problem into a set of convex optimization subproblems, we can guarantee the convergence of our proposed algorithm. Additionally, and in contrast to Analysis K-SVD, our proposed method can naturally adapt in representing data of varying underlying sparsity with no prior knowledge.

3. From SNS to ADL. In this section, we reformulate the SNS problem and the ADL problem in a matrix form, and then establish their equivalence.

Given any $m \times n$ matrix $\mathbf{A}$ such that $\text{row}(\mathbf{A}) \subset R^n$, the SNS problem may be defined as follows,

$$SNS(\mathbf{A}) = \arg\min_{\mathbf{N}} \|\mathbf{N}\|_0, \ s.t. \ \text{col}(\mathbf{N}) = \text{null}(\mathbf{A}). \tag{3.1}$$

Let $\mathbf{X} = [x_1, \ldots, x_n]$ be a generic data matrix, the ADL problem can then be rewritten as

$$ADL(\mathbf{X}) = \arg\min_{\mathbf{U}} \|\mathbf{U}\|_0, \ s.t. \ \mathbf{DX} = \mathbf{U}, \text{row}(\mathbf{X}) = \text{row}(\mathbf{U}), \tag{3.2}$$

where $\mathbf{D}$ is an analysis operator in matrix form, and $\mathbf{U}$ is the associated sparse coefficient matrix. To avoid a trivial solution such as $\mathbf{U} = 0$, we further require $\text{row}(\mathbf{X}) = \text{row}(\mathbf{U})$. Essentially, this is the maximum information we can preserve for $\mathbf{X}$, since all rows of $\mathbf{U}$ is a linear combination of rows in $\mathbf{X}$, and hence $\text{row}(\mathbf{U}) \subseteq \text{row}(\mathbf{X})$. In practice, we may also consider the case that $\text{row}(\mathbf{U}) \subset \text{row}(\mathbf{X})$ by further selecting a subset of d_i in $\mathbf{D}$. We are focusing here on the generic formulation, i.e. $\text{row}(\mathbf{X}) = \text{row}(\mathbf{U})$, for the sake of theoretical analysis, and will elaborate on this issue later in the detailed discussion of the algorithm.

We note that finding a sparse representation of the null space in Problem 3.1, is equivalent to sparsifying a given matrix $\hat{\mathbf{N}}$ such that $\text{col}(\hat{\mathbf{N}}) = \text{null}(\mathbf{A})$. This coincides with the goal of Problem 3.2 where the row space of data matrix $\mathbf{X}$ is instead invoked. In particular, we formally have the following result,

THEOREM 3.1. *Assume* $\text{null}(\mathbf{A}) = \text{row}(\mathbf{X})$*, then a matrix* $\mathbf{N}$ *is a minimizer of the SNS problem*

(as shown in Eq. (3.1)), if and only if $\mathbf{N}^T$ *is a minimizer of the ADL problem (as shown in Eq. (3.2))* .

Proof. Assume $\mathbf{N}$ is a minimizer of (3.1), then the constraints in (3.1) ensure that $\text{null}(\mathbf{A}) = \text{col}(\mathbf{N})$. Since $\text{col}(\mathbf{X}^T) = \text{row}(\mathbf{X}) = \text{null}(\mathbf{A})$, we then have

$$\text{col}(\mathbf{X}^T) = \text{col}(\mathbf{N}). \tag{3.3}$$

We next consider any optimizer $\mathbf{U}$ of problem (3.2). Since $\text{row}(\mathbf{X}) = \text{row}(\mathbf{U})$, when combined with the condition $\text{row}(\mathbf{X}) = \text{null}(\mathbf{A})$, we have $\text{row}(\mathbf{U}) = \text{row}(\mathbf{X}) = \text{row}(\mathbf{N}^T)$. $\mathbf{N}^T$ is therefore a feasible solution of Eq. (3.2), and so is $\mathbf{U}^T$ of Eq. (3.1). It follows that $\|\mathbf{U}\|_0 = \|\mathbf{N}^T\|_0$, and hence $\mathbf{N}^T$ is also a minimizer of (3.2) and as is $\mathbf{U}^T$ for Eq. (3.1). ∎

This essentially tells us that we can solve a sparse dictionary learning problem, should we have access to an effective method of solving the corresponding SNS problem. Specifically, given a data matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, the analysis dictionary for $\mathbf{X}$ may be constructed in the following three steps:

1. Construct a matrix $\mathbf{A}$ such that $\text{row}(\mathbf{A}) = \text{null}(\mathbf{X})$, i.e. $\mathbf{X}\mathbf{A}^T = 0$ and $rank(\mathbf{A}) + rank(\mathbf{X}) = n$.
2. Find the sparse feature vectors $\mathbf{U}^T$ by solving $\mathbf{N} = SNS(\mathbf{A})$.
3. Construct the analysis operator $\mathbf{D}$ from $\mathbf{D}\mathbf{X} = \mathbf{U}$.

4. An iterative sparse null space pursuit. We have discussed the relation of SNS and ADL in Section 3, and have shown that they may be cast in one unified framework of sparse null space pursuit. Nevertheless, solving SNS is itself a difficult problem. Coleman and Pothen [8] have proved that SNS is essentially NP-hard, hence ruling out a polynomial time algorithm. We, however, show that it is still possible to approximate the sparse null space basis in polynomial time. In this section, we propose an l_1-based iterative optimization method for a sparse null space pursuit.

4.1. A greedy algorithm for the SNS problem. Previous works on the SNS problem have shed some light towards a solution in polynomial time. In [8], the authors proposed a greedy algorithm for the SNS problem. For the sake of clarity and further discussion, we refer to it as Algorithm 1. Additionally, it has been proved in [8] that Algorithm 1 can be used to construct a sparse null space basis, as stated in Theorem 4.1 [8].

Algorithm 1 A greedy algorithm for sparse null space problem

Initialize: matrix $\mathbf{A} \in R^{m \times d}$, $\mathbf{N} = \emptyset$
for i = 1, ... , q **do**
  Find a sparsest null vector $\mathbf{n}_i$ such that $rank(\mathbf{N} \oplus \mathbf{n}_i) = i$.
  $\mathbf{N} = \mathbf{N} \oplus \mathbf{n}_i$
end for

THEOREM 4.1. *A matrix* $\mathbf{N}$ *is a sparsest null basis of* $\mathbf{A}$ *if and only if it can be constructed by the greedy algorithm.*

It is worth noting that the maximum number of iterations q in Algorithm 1 is constrained by the rank of $\mathbf{A}$, i.e. $q = d - rank(\mathbf{A})$. Moreover, this greedy algorithm can find the global optimal solution for the SNS problem. This elegant result amounts to finding the sparsest null space basis of $\mathbf{A}$ in exactly q steps. The subproblem of finding a sparsest null vector itself is, however, also

a NP-hard problem [8]. We therefore next focus on finding a method to solve this subproblem in each iteration of Algorithm 1.

4.2. l_1-based search for sparse null space. We first reformulate the subproblem of finding a sparsest null vector in Algorithm 1 as follows,

$$\begin{aligned} &\min_{\mathbf{n}_i} \|\mathbf{n}_i\|_0, \\ &s.t.\ \mathbf{A}\mathbf{n}_i = 0, P_{\mathbf{N}_{i-1}^{\perp}}\mathbf{n}_i \neq 0, \end{aligned} \tag{4.1}$$

where $\mathbf{N}_{i-1}$ is the subspace spanned by the constructed null space vectors in the previous $(i-1)^{th}$ iteration. The condition $P_{\mathbf{N}_{i-1}^{\perp}}\mathbf{n}_i \neq 0$ implies that $\mathbf{n}_i$ is not in the current span of $\mathbf{N}$, and hence $rank(\mathbf{N}_i \oplus \mathbf{n}_i) = rank(\mathbf{N}_i) + 1$.

There are two inherent difficulties in this formulation. First, $\|\cdot\|_0$ is of combinatorial nature, hence the reason of the NP-hardness of the problem. Second, the constraint in Eq. (4.1)

$$P_{\mathbf{N}_i^{\perp}}\mathbf{n}_i \neq 0, \tag{4.2}$$

defines a region that is neither compact nor convex. To address the first problem, we propose to take advantage of established results on sparsity pursuit via l_1 minimization [6][7]. To address the second one, and to hence obtain a convex and compact feasible region, we propose to restate the condition $P_{\mathbf{N}_i^{\perp}}\mathbf{n}_i \neq 0$ as follows,

$$\exists j \in \{1, \ldots, d\}, (P_{\mathbf{N}_{i-1}^{\perp}}\mathbf{n}_i)_j = c, \tag{4.3}$$

where c is a positive constant.

Additionally, we establish the following lemma to justify the variation on the constraint from (4.2) to (4.3):

LEMMA 4.2. *The solution of Problem 4.1 remains invariant if the constraint (4.2) is substituted by constraint (4.3).* The proof of Lemma 4.2 is presented in Appendix A.

The meaning of Lemma 4.2 is that we may then separate the region defined by (4.3) into compact and convex regions based on j, i.e. the location of the forced nonzero element. Since the optimal solution must reside in one of these regions, we may search for the sparsest null vector in each region from $j = 1$ to d. We subsequently have a convex formulation of l_1 minimization for each j. Algorithm 1 may then be specifically realized as Algorithm 2.

Algorithm 2 Sparse Null Space Basis Pursuit

```
Initialize: matrix A, N = ∅
for i = 1, ... , p do
    for j = 1, ... , d do
        Find n_i^j = arg min ||n||_1, s.t. An = 0, (P_{N^⊥} n)_j = c
    end for
    n_i = arg min ||n_i^j||_0
    N = N ⊕ n_i
end for
```

This is tantamount to solving the following optimization problem for each j in Algorithm 2,

$$\begin{aligned} &\min_{\mathbf{n}} \|\mathbf{n}\|_1, \\ &s.t.\ \mathbf{An} = 0, (P_{\mathbf{N}^\perp}\mathbf{n})_j = c. \end{aligned} \tag{4.4}$$

It is worth noting that the exact recovery of each $\mathbf{n}$ via Eq. (4.4) is determined by the incoherence of the linear operator defined by the two constraints and the sparsity of each $\mathbf{n}$. To solve (4.4), we adopt the framework of augmented Lagrange method (ALM) on account of its superior performance matrix-norm minimization problems [17] [18]. Specifically, we have the augmented Lagrange function of (4.4) as

$$\begin{aligned} L(\mathbf{n}, \mathbf{Y}_1, \mathbf{Y}_2, \mu) = &\|\mathbf{n}\|_1 + \langle \mathbf{Y}_1, \mathbf{An}\rangle + \langle \mathbf{Y}_2, (P_{\mathbf{N}^\perp}\mathbf{n})_j - c\rangle \\ &+ \frac{\mu}{2}\|\mathbf{An}\|^2 + \frac{\mu}{2}\|(P_{\mathbf{N}^\perp}\mathbf{n})_j - c\|^2. \end{aligned} \tag{4.5}$$

The primal variable $\mathbf{n}$ is first updated in each iteration with fixed dual variables Y_1, Y_2 and μ. By introducing an auxiliary variable η, we have

$$\mathbf{n}_{k+1} = \mathcal{T}_{\frac{1}{\mu_k \eta}}\left(\mathbf{n}_k - \frac{\mathbf{n}_k^1 + \mathbf{n}_k^2}{\eta}\right), \tag{4.6}$$

where $\mathcal{T}$ is the soft-thresholding operator, and $\|\eta\|^2 \geq \|\mathbf{A}\|^2 + \|P_{\mathbf{N}^\perp}\|^2$, [1]and

$$\mathbf{n}_k^1 = \mathbf{A}^T\left(\mathbf{An}_k + \frac{\mathbf{Y}_1^k}{\mu_k}\right), \tag{4.7}$$

$$\mathbf{n}_k^2 = P_{\mathbf{N}^\perp}\left((P_{\mathbf{N}^\perp}\mathbf{n})_j - c + \frac{\mathbf{Y}_2^k}{\mu_k}\right)_j^*. \tag{4.8}$$

Next, the dual variables $\mathbf{Y}_1$, $\mathbf{Y}_2$ and μ are updated as

$$\mathbf{Y}_1^{k+1} = \mathbf{Y}_1^k + \mu_k\left(\mathbf{An}_{k+1}\right), \tag{4.9}$$

$$\mathbf{Y}_1^{k+1} = \mathbf{Y}_1^k + \mu_k\left((P_{\mathbf{N}^\perp}\mathbf{n})_j - c\right), \tag{4.10}$$

$$\mu_{k+1} = \min\{\rho\mu_k, \mu_{max}\}. \tag{4.11}$$

The strategy of linearized ALM method provides a fast convergence rate [18]. This effectively provides us a method (Algorithm 2), named Sparse Null Space Basis Pursuit (SNS-BP) in this paper, to solve the SNS problem efficiently.

4.3. Solving the ADL problem via sparse null space basis pursuit. In Section 3, we discussed the equivalence of the ADL problem and the SNS problem, and hence further describe the details of solving the ADL problem (as in (3.2)) via SNS-BP.

For a typical ADL problem as in Eq. (3.2), the first step, as discussed in Section 3, is to construct a matrix $\mathbf{A}$ whose transpose is the null space of $\mathbf{X}$. Concretely, we have the following problem,

PROBLEM 1. *Find* $\mathbf{A}$ *such that* $\mathbf{XA}^T = 0$.

[1]The value of η is selected in this way to insure the convergence of the algorithm. The details are discussed in [18].

A simple way would be to consider a singular value decomposition of $\mathbf{X}$, and keep the right singular vectors with zero singular values coinciding with the rows of $\mathbf{A}$. For a common scenario where the data matrix $\mathbf{X}$ is contaminated by Gaussian noise, we can set $\mathbf{A}$ to the right singular vectors with small singular values, instead of exactly zero. This in fact offers an additional advantage of filtering out dense Gaussian noise from the data matrix $\mathbf{X}$.

Upon constructing $\mathbf{A}$, we proceed to obtain the sparse coefficient matrix $\mathbf{U}^T = SNS(\mathbf{A})$. Note that $\mathbf{U}$ is computed independently of the analysis operator $\mathbf{D}$. In case $\mathbf{D}$ is required for further processing incoming data, using $\mathbf{DX} = \mathbf{U}$, we may easily obtain $\mathbf{D} = \mathbf{UX}^\dagger$, where $\mathbf{X}^\dagger$ is the pseudo-inverse of $\mathbf{X}$. In particular, if all entries of the dataset are all independent, i.e. $\mathbf{X}$ is full row rank, then $\mathbf{X}^\dagger = \mathbf{X}^T(\mathbf{XX}^T)^{-1}$.

While we formulated the ADL problem with the constraint $\text{row}(\mathbf{U}) = \text{row}(\mathbf{X})$ in Section 3, if a more compact representation of $\mathbf{X}$ is preferred, we may opt for $\text{row}(\mathbf{U}) \subset \text{row}(\mathbf{X})$, hence further reducing the dimension of the original data space.

5. Numerical experiments on SNS and ADL. For a quantitative evaluation of our algorithm, we synthesize data that is compatible with the model of SNS and ADL, and show that SNS-BP is able to reconstruct the sparse null space basis of the SNS problem/the sparse coefficients of the ADL problem.

First, we synthesize a $d \times q$ sparse matrix $\mathbf{N}$ as the sparse null space basis of some matrix $\mathbf{A}$, where $\mathbf{A}$ may be constructed by exploiting the SVD decomposition of $\mathbf{N}$, and by selecting its left singular vectors corresponding to the zero singular values as the rows of $\mathbf{A}$, i.e. $\mathbf{AN} = \mathbf{0}$.

All elements in $\mathbf{N}$ follow a binomial distribution as zero/nonzero entries. Moreover, the amplitude of each nonzero element is generated from a gaussian distribution.

The matrix $\mathbf{A}$ can then be seen as the input to SNS-BP, and we may therefore compare the recovered nullspace basis $\hat{\mathbf{N}}$ with the ground truth $\mathbf{N}$. In Fig.1, we show one example of exact recovery of a sparse nullspace basis up to permutation and scale.

(a) Randomly generated sparse basis $\mathbf{N}$

(b) Recovered sparse nullspace basis $\hat{\mathbf{N}}$

FIG. 1. *An example of the result of SNS-BP*

In Fig.2, we present the sparsity level of $\hat{\mathbf{N}}$ with the sparsity of $\mathbf{N}$ varying from 0.01 to 0.2, i.e. 1% nonzero to 20% nonzero. If our method works well, we would expect it to find the sparsest basis, and therefore $\rho(\hat{\mathbf{N}}) \approx \rho(\mathbf{N})$, i.e. the relative sparsity $\rho(\hat{\mathbf{N}})/\rho(\mathbf{N}) \approx 1$. In Fig.2, 10 experiments have been carried out and the average sparsity is calculated. We can see that the sparse bases discovered by SNS-BP have similar sparsity with $\mathbf{N}$, with $\rho(\mathbf{N})$ varying from 0.01 to 0.2. Additionally, we define the relative error of $\hat{\mathbf{N}}$ as

$$err(\hat{\mathbf{N}}) = \frac{\|\hat{\mathbf{N}}\mathbf{P}\mathbf{\Gamma} - \mathbf{N}\|_F}{\|\mathbf{N}\|_F}, \tag{5.1}$$

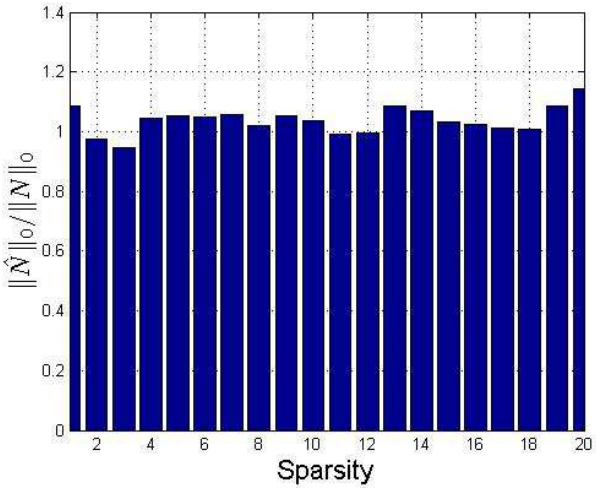
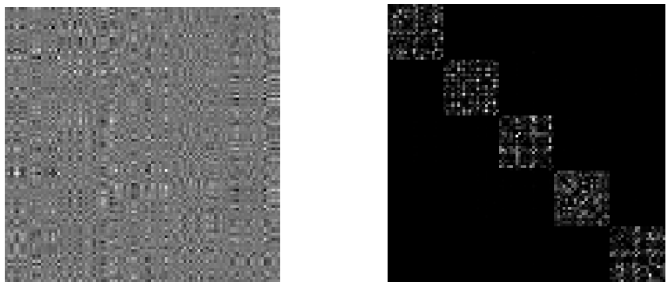


FIG. 2. $\|\hat{\mathbf{N}}\|_0/\|\mathbf{N}\|_0$ *vs Sparsity*

where $\mathbf{P}$ is an arbitrary permutation matrix, and $\mathbf{\Gamma}$ is a diagonal matrix representing the scales of each sparse basis. The average relative error of all the experiments with the sparsity of $\mathbf{N}$ varying from 0.01 to 0.2, is 1.69%.



(a) Synthetic Data matrix $\mathbf{X}$ (b) Sparse coefficient matrix W

FIG. 3. *Sample synthetic data matrix and its underlying structure*

We next test the analysis dictionary learning via SNS-BP by exploring data samples with hidden underlying sparse structures. In particular, data samples are randomly selected from a union of low-dimensional subspaces $S = S_1 \cup S_2 \cdots$, in which each subspace is also randomly constructed by using the orthogonal basis of a set of uniformly distributed vectors. Under this setting, each sample can be represented as a linear combination of other samples in the same subspace. The dataset, written as a matrix $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_n}]$ as shown in Fig.3(a), has a sparse intrinsic structure W such that $\mathbf{X} = \mathbf{XW}$, where W is a block-diagonal matrix as Fig.3(b), and each block represents one subspace. In our experiment, we have data points distributed in five 3-dimensional subspaces within the ambient space R^{100}. It hence implies that the nullspace of $\mathbf{A}$ constructed from $\mathbf{X}$ is of dimension 15. An analysis dictionary $\mathbf{D}$ is then trained using SNS-BP, and the associated sparse

coefficient matrix $\mathbf{U}$ is obtained as shown in Fig.4(a). Specifically, we can see that all nonzero entries of each sparse vector are clustered together, corresponding to data samples from the same subspace. In other words, for an atom d_i in the analysis dictionary $\mathbf{D}$, all the data points that have a significant response to it, are from one subspace, and the rest of the data have zero response.

(a) The sparse coefficient matrix $\mathbf{U}$ by ADL using SNS-BP

(b) Permute rows of $\mathbf{U}$ to show the structure of $\mathbf{X}$

FIG. 4. *Sparse coefficients using learned analysis dictionary*

We next cluster the rows of $\mathbf{U}$ and permute them accordingly, as is presented in Fig.4(b). It is interesting to see that a more compact block-diagonal structure emerges again. Note that in this example we find in totality 15 atoms, with three atom sets each supporting the data samples (have largely nonzero inner product) in a subspace. This number corresponds to the intrinsic dimension of each subspace. Having trained $\mathbf{D}$, it is subsequently simple to figure out which subspace a data point belongs to: we may simply separate $\mathbf{D}$ into $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \cdots]$, with $\mathbf{D}_i$ being the set of atoms supporting the ith subspace, and ultimately determine the maximal $\|\mathbf{D}_i\mathbf{x}\|$. It is therefore more efficient to recover the underlying structure of a given dataset and represent this structure in a more compact way.

5.1. In Relation to Analysis K-SVD. Analysis K-SVD arguably provides the state-of-the-art solution for the ADL problem, and has achieved a much improved performance in signal denoising by discovering the underlying UoS structure of a wide class of signals [25]. We next comparatively evaluate the performance of SNS-BP and Analysis K-SVD in the recovery problem of UoS of synthesized data.

The synthesis of data from a union of subspaces is similar to the setting in Section 5. In particular, data samples are randomly chosen from a union of low-dimensional subspaces $S = S_1 \cup S_2 \cdots$, in which each subspace is also randomly constructed by using the orthogonal basis of a set of uniformly distributed vectors. In generating our data, we pick an ambient space of dimension 100, and collectively 5 subspaces. We evaluate the performance of SNS-BP and Analysis K-SVD by varying the intrinsic dimensions of each subspace. Interestingly, SNS-BP shows constant performance with the increase of the intrinsic dimension of each subspace, while the performance of Analysis K-SVD deteriorates when the intrinsic dimension of each subspace increases.

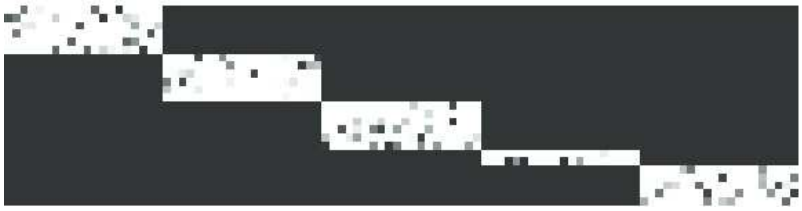
(a) The sparse coefficient matrix $\mathbf{U}$ by ADL using SNS-BP

(b) The sparse coefficient matrix $\mathbf{U}$ by ADL using Analysis K-SVD

FIG. 5. *Sparse coefficients for 5 subspaces with intrinsic dimension 3*

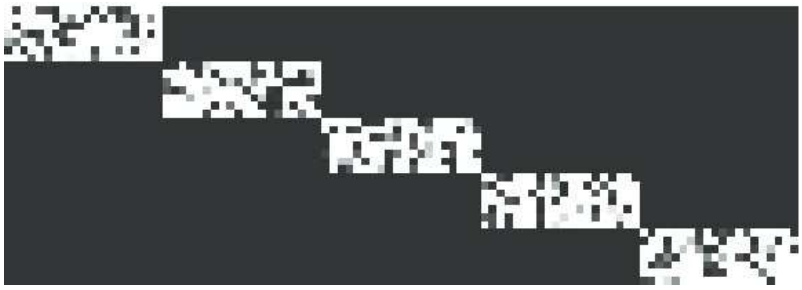
As shown in Figs. 5 and 6, when the intrinsic dimension is low, both SNS-BP and Analysis K-SVD can effectively learn atoms from the dataset which yield sparse coefficients to efficiently reflect the underlying subspace of each data point. More precisely, each atom has a strong response on data in only one subspace. and is rather absent in other subspaces. Five blocks corresponding to five subspaces appear in the sparse coefficient matrix using both approaches. SNS-BP exhibits a clearly improved performance over Analysis K-SVD, in recovering the intrinsic dimension of each subspace, as shown in Fig. 5 where the intrinsic dimension of each subspace is 3, and in Fig. 6 as the dimension increases to 5. This may be understood by recalling that the algorithm SNS-BP ensures that the sparse vector determined at each iteration is orthogonal to the subspace spanned by the previously found sparse vectors (i.e. truly novel). Equivalently said, the rows in Fig. 5(a)/Fig. 6(a) are linearly independent- a property whichh provides an accurate estimate of the intrinsic dimension of each subspace. In contrast, since Analysis K-SVD can possibly find collinear sparse vectors, it yield more sparse vectors than the dimension of a given subspace, and may hence partially miss another subspace.

(a) The sparse coefficient matrix $\mathbf{U}$ by ADL using SNS-BP

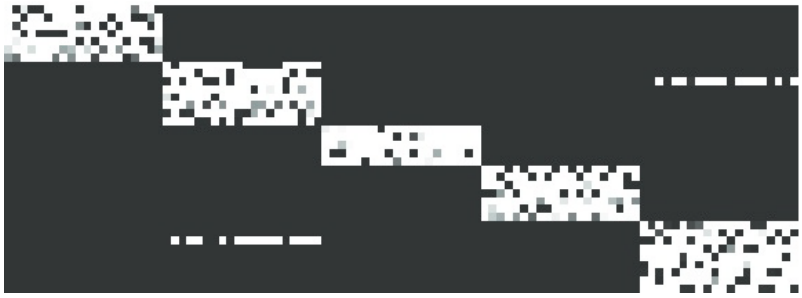


(b) The sparse coefficient matrix $\mathbf{U}$ by ADL using Analysis K-SVD

FIG. 6. *Sparse coefficients for 5 subspaces with intrinsic dimension 5*



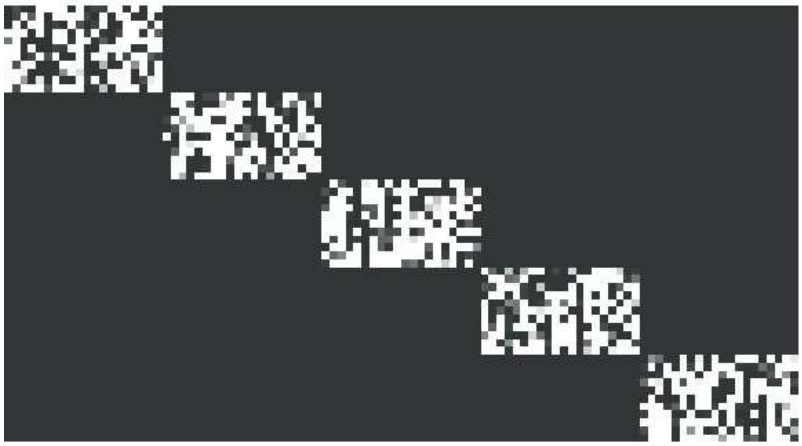
(a) The sparse coefficient matrix $\mathbf{U}$ by ADL using SNS-BP



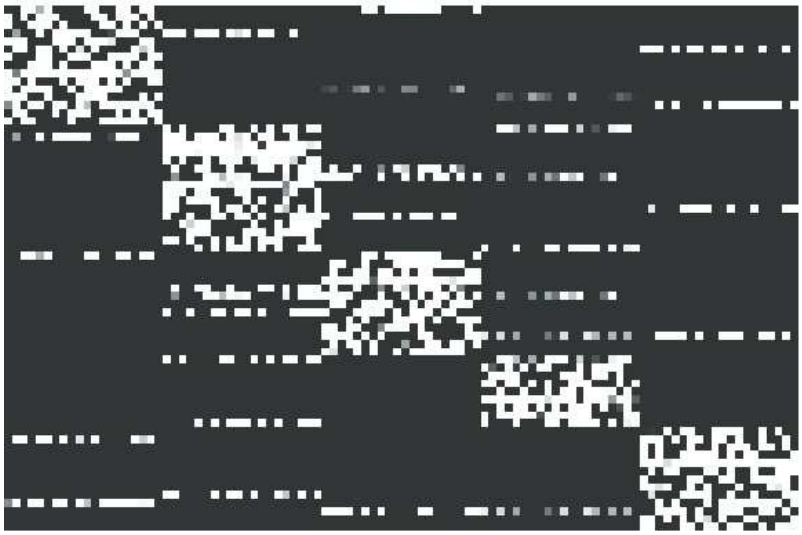
(b) The sparse coefficient matrix $\mathbf{U}$ by ADL using Analysis K-SVD

FIG. 7. *Sparse coefficients for 5 subspaces with intrinsic dimension 7*

In Figs. 7, 8, 9 and 10, the intrinsic dimension of each subspace is 7, 11, 13 and 15, respectively. As the intrinsic dimension increases, Analysis K-SVD begins to learn atoms which span two or more subspaces. Such cases become more prevalent as the intrinsic dimension of each subspace increases. In Fig. 10 for instance, the 5 coefficient blocks representing the 5 subspaces merge and display significant responses from two or more subspaces. SNS-BP, in contrast, exhibits a more consistent performance even as the intrinsic dimension of each subspace increases, thus resulting in recovering a correct dimension for each subspace.

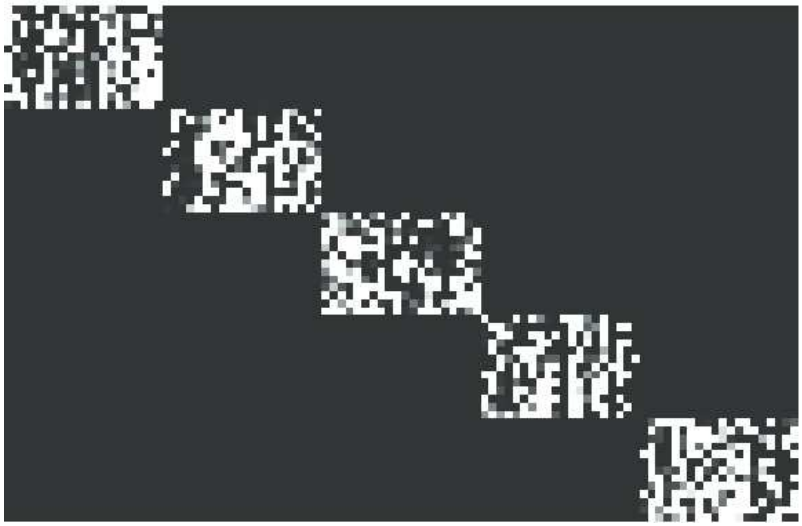


(a) The sparse coefficient matrix $\mathbf{U}$ by ADL using SNS-BP

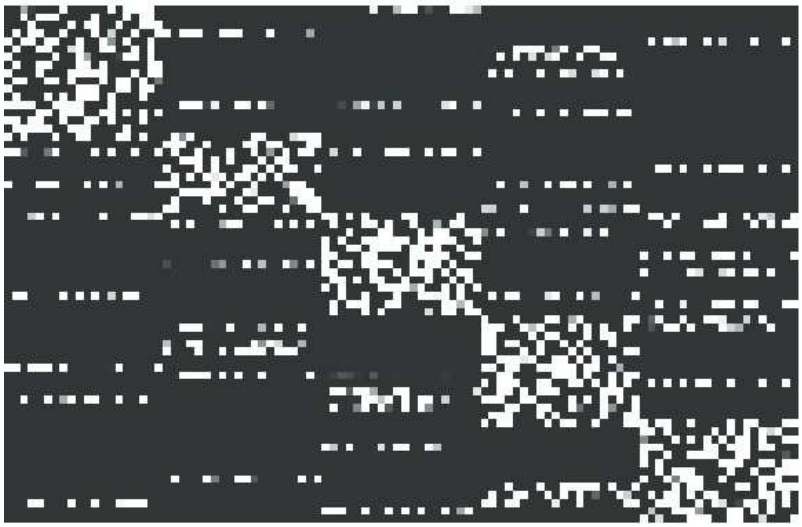


(b) The sparse coefficient matrix $\mathbf{U}$ by ADL using Analysis K-SVD

FIG. 8. *Sparse coefficients for 5 subspaces with intrinsic dimension 11*

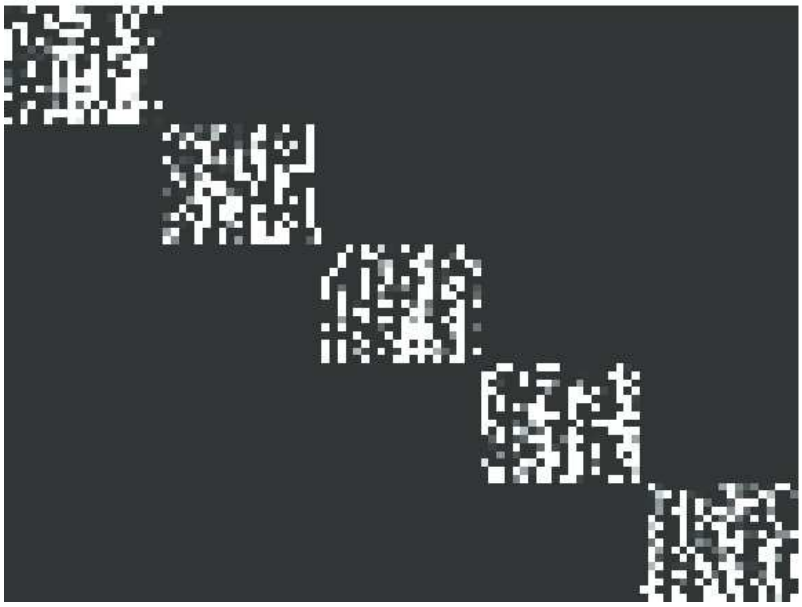


(a) The sparse coefficient matrix $\mathbf{U}$ by ADL using SNS-BP

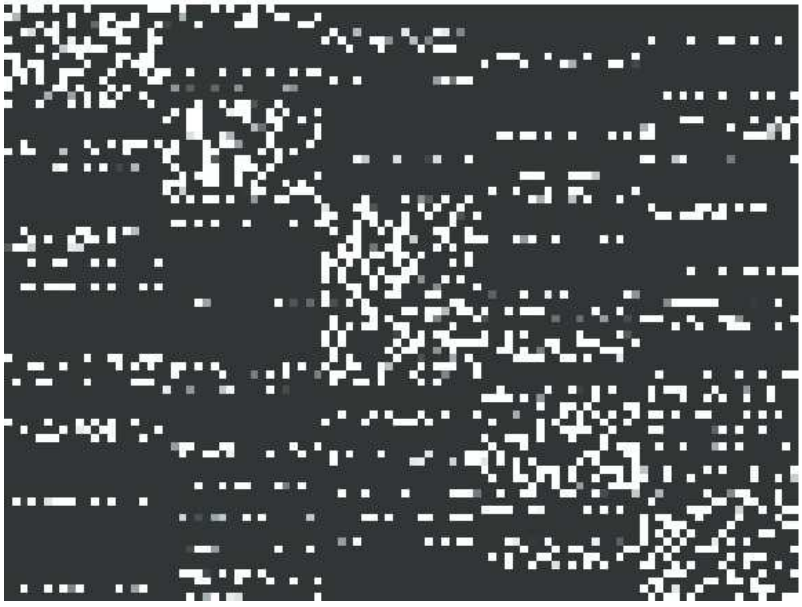


(b) The sparse coefficient matrix $\mathbf{U}$ by ADL using Analysis K-SVD

FIG. 9. *Sparse coefficients for 5 subspaces with intrinsic dimension 13*



(a) The sparse coefficient matrix $\mathbf{U}$ by ADL using SNS-BP

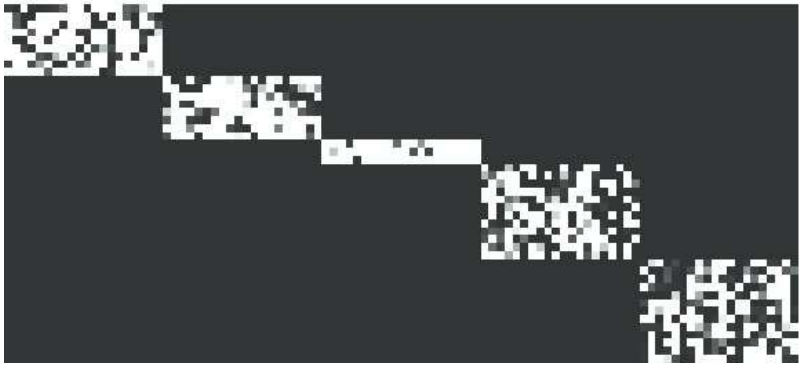


(b) The sparse coefficient matrix $\mathbf{U}$ by ADL using Analysis K-SVD

FIG. 10. *Sparse coefficients for 5 subspaces with intrinsic dimension 15*

In Fig. 11, we show a case of UoS with different randomly generated dimensions. It worth noting that given the ambient space of R^100, we limit the intrinsic dimension of each subspace between 1 and 19, to avoid the extreme case that when every subspace is of dimension 20, then we

may run into the situation that all samples are randomly chosen from R^100 and lose any non-trivial UoS structure. In this case, Analysis K-SVD misses one subspace with the lowest dimension entirely after learneding 45 atoms, and a non-negligible portion of the dictionary fails to distinguish data from different subspaces. SNS-BP shows a clear 5-subspace of the same dataset.



(a) The sparse coefficient matrix $\mathbf{U}$ by ADL using SNS-BP

(b) The sparse coefficient matrix $\mathbf{U}$ by ADL using Analysis K-SVD

FIG. 11. *Sparse coefficients for 5 subspaces with intrinsic dimension randomly picked between 1 and 19 as 9, 8, 3, 12, 13*

We further thoroughly test the performance of SNS-BP and Analysis K-SVD on a dataset with intrinsic dimension from 5 to 16 using the measure of intra-cluster covariance ratio. For each dimension, we repeat the process of data generation 20 times, to obtain the mean value of the intra-cluster correlation ratios for both methods.

We define the intra-cluster covariance ratio as

$$CV(\mathbf{U}) = \frac{\text{Intra-cluster covariance of } \mathbf{U}}{\text{total covariance of } \mathbf{U}}. \tag{5.2}$$

In Fig. 12, we can see that the CV of Analysis K-SVD deteriorates from 1 to around 0.75 with the increase of the intrinsic dimension of each subspace, While SNS-BP maintains a consistently high performance throughout.

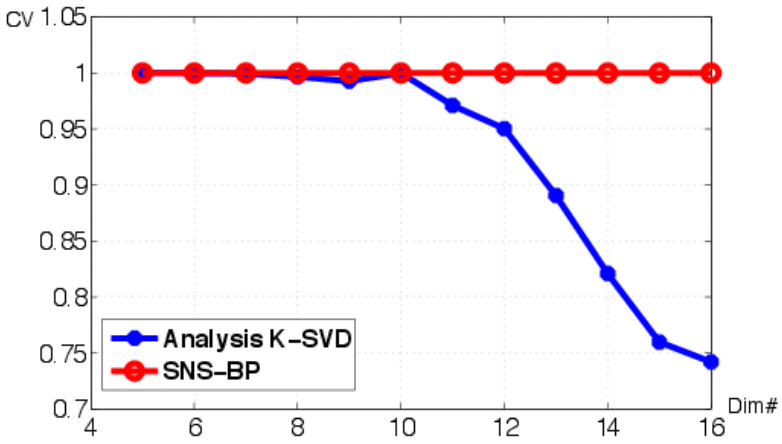


FIG. 12. *Intra-cluster covariance ratio for SNS-BP and Analysis K-SVD. The x-axis is the intrinsic dimension of each subspace, and the y-axis is the CV value.*

We further test the performance of SNS-BP and Analysis K-SVD by randomly choosing the intrinsic dimension of each subspace between 1 and 19. In this experiment, the intrinsic dimension of each subspace can be different. We repeat the data generation 100 times to obtain mean CV values for both methods. The results show a higher CV value 1.00 of SNS-BP versus 0.97 of Analysis K-SVD.

All these experiments in this part prove a more robust performance of SNS-BP compared to Analysis K-SVD when the intrinsic dimension of each subspace is higher. Moreover, SNS-BP demonstrates an ability of recovering the intrinsic dimensionality of each underlying subspace, and of automatically avoiding redundant atoms.

5.2. Applications on real-world data. In this experimental section, we explore the inference potential of our method on images. The performance of our algorithm is evaluated on texture images from the Brodatz database [22]. Each texture image is partitioned into a set of patches, and their analysis operator is learned from patches of different textures. The latter is subsequently applied to incoming data, which is also segmented into sets of patches. The properties of various textures may lead to different patterns of the corresponding sparse coefficients. For example, the texture with more randomness may lead to less sparsely structured coefficients and more coherent/correlated textures, i.e. on account of a broadly spread distribution of its patches, discovered upon applying the learned operator to incoming data. Note that we avoid the complexity of processing the order of patches by considering the distribution of the coefficients instead of matching the output vectors.

Specifically, we segment each texture image into 10×10 patches, and randomly pick a subset of 120 patches as the training set from each texture image, and the rest of the patches are used as a testing set. The texture images from the Brodatz database [22], and the corresponding sample patches are shown in Fig. 13. In our experiment, we first train the analysis operator by using half of the data in the training set without knowing the label of each patch, and then calculate the distribution of the coefficients P_i of the rest of the patches from the i^{th} class of texture in the training set. In the next testing stage, texture images are used as a set of patches, for which the

distribution of the coefficients $\mathbf{U}_j = \mathbf{D}\mathbf{X}_j$ with all P_i are compated. In particular, we assign $\mathbf{X}_j$ to the class with the closest distribution, such as

$$class(\mathbf{X}_j) = \arg\min_i d(P_i, P_{\mathbf{U}_j}). \tag{5.3}$$

We use the total variation distance in ((5.3), as defined in [16],

$$d(p,q) = \|p-q\|_{TV} = \frac{1}{2}\sum_{x\in\Omega} |p(x) - q(x)|. \tag{5.4}$$

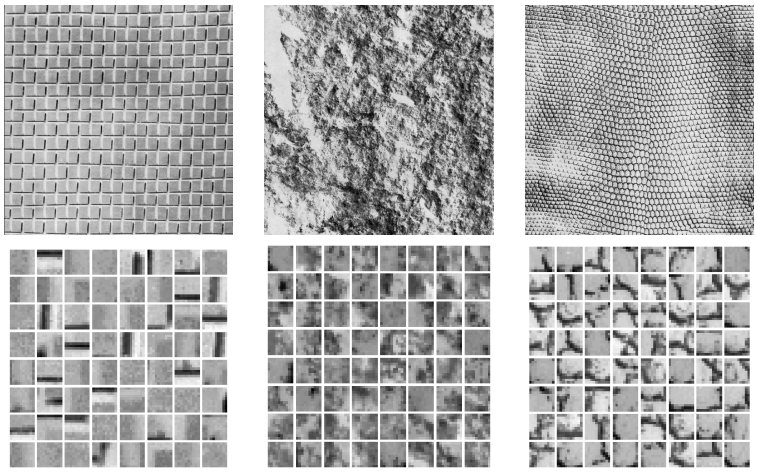


FIG. 13. *Textures and the corresponding patches for training*

In this experiment, the classification rate is 97.78% for the texture images shown in Fig. 13. The performance is higher than known state of the art methods based on predesigned features, such as [26] with a 86.63% classification rate, and comparable to the supervised dictionary learning algorithm [21]. It isimportant to note that, the training set in our case, is only composed of around 1% of the dataset. A less stringent training set implies a lower computational cost. This also demonstrates the scalability of our method, in light of its competitive classification performance.

6. Conclusion. We have proposed in this paper, a novel approach for the sparse nullspace problem, and have unveiled its equivalence to the ADL problem. We have presented the SNS-BP, an iterative algorithm based on l_1 minimization, to pursue the solution of the SNS problem. We have further applied this algorithm to analysis dictionary learning, and show the efficacy of our approach by experiments on both synthetic dataset and real-world data in texture classification.

Future work may include several aspects related to both SNS problem and ADL. The relation between SNS and nonlinear dimension reduction needs further investigation and may lead to results on graph embedding. Moreover, our future goal is to explore the potential application of ADL on other high dimensional database, such as image/video classification.

Appendix. Proof of Lemma 4.2.

Lemma 4.2 states that the following optimization problems are equivalent.

$$\min_{\mathbf{n}} \|\mathbf{n}\|_0, \quad s.t.\ \mathbf{An} = 0, P_{\mathbf{N}^\perp}\mathbf{n} \neq 0, \tag{A}$$

$$\min_{\mathbf{n}} \|\mathbf{n}\|_0, \quad s.t.\ \mathbf{An} = 0, \exists j \in \{1, \ldots, d\}, (P_{N^\perp}\mathbf{n})_j = c. \tag{B}$$

Proof. First, we show that if $\mathbf{n}$ is an optimal solution of (A), then for some real number α, $\alpha\mathbf{n}$ is also a minimizer of (B).

Any minimizer $\mathbf{n}'$ of (B), also satisfies the constraints of (A). It hence follows that

$$\|\mathbf{n}'\|_0 \geq \|\mathbf{n}\|_0. \tag{A.1}$$

Assume $|(P_{\mathbf{N}^\perp}\mathbf{n})_k| = \|P_{\mathbf{N}^\perp}\mathbf{n}\|_\infty$, and noting that $\|P_{\mathbf{N}^\perp}\mathbf{n}\|_\infty \neq 0$, we construct

$$\hat{\mathbf{n}} = \frac{c}{(P_{\mathbf{N}^\perp}\mathbf{n})_k} \cdot \mathbf{n} = \alpha\mathbf{n}. \tag{A.2}$$

Since $(P_{\mathbf{N}^\perp}\hat{\mathbf{n}})_k = c$, $\hat{\mathbf{n}}$ is also feasible in (B). Suppose that $\|\hat{\mathbf{n}}\|_0 = \|\mathbf{n}\|_0 \leq \|\mathbf{n}'\|_0$, and in combination of Eq. A.1, we can conclude that $\hat{\mathbf{n}} = \alpha\mathbf{n}$, and is also a solution of (B), and therefore $\|\hat{\mathbf{n}}\|_0 = \|\mathbf{n}'\|_0$.

Then it is trivial to show that $\mathbf{n}'$ is also a minimizer of (A), given the fact that $\mathbf{n}'$ is a feasible solution of (A) and $\|\mathbf{n}'\|_0 = \|\hat{\mathbf{n}}\|_0 = \|\mathbf{n}\|_0$, thus proving Lemma 4.2. □

REFERENCES

[1] Michal Aharon, Michael Elad, and Alfred Bruckstein, *K-svd: An algorithm for designing overcomplete dictionaries for sparse representation*, Signal Processing, IEEE Transactions on, 54 (2006), pp. 4311–4322.

[2] Xiao Bian and Hamid Krim, *Optimal operator space pursuit: a framework for video sequence data analysis*, in Computer Vision–ACCV 2012, Springer, 2013, pp. 760–769.

[3] Xiao Bian and Hamid Krim, *Robust Subspace Recovery via Bi-Sparsity Pursuit*, ArXiv e-prints, (2014).

[4] Olivier Bousquet and Léon Bottou, *The tradeoffs of large scale learning*, in Advances in neural information processing systems, 2008, pp. 161–168.

[5] Emmanuel J Candes, *The restricted isometry property and its implications for compressed sensing*, Comptes Rendus Mathematique, 346 (2008), pp. 589–592.

[6] Emmanuel J. Candès, Justin K. Romberg, and Terence Tao, *Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information.*, IEEE Transactions on Information Theory, 52 (2006), pp. 489–509.

[7] Emmanuel J Candès, Justin K Romberg, and Terence Tao, *Stable signal recovery from incomplete and inaccurate measurements*, Communications on pure and applied mathematics, 59 (2006), pp. 1207–1223.

[8] Thomas F Coleman and Alex Pothen, *The null space problem i. complexity*, SIAM Journal on Algebraic Discrete Methods, 7 (1986), pp. 527–537.

[9] David L Donoho and Carrie Grimes, *Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data*, Proceedings of the National Academy of Sciences, 100 (2003), pp. 5591–5596.

[10] Michael Elad, *Sparse and redundant representation modeling: What next?*, IEEE Signal Processing Letters, 19 (2012), pp. 922–928.

[11] Michael Elad and Michal Aharon, *Image denoising via sparse and redundant representations over learned dictionaries*, Image Processing, IEEE Transactions on, 15 (2006), pp. 3736–3745.

[12] Michael Elad, Mario AT Figueiredo, and Yi Ma, *On the role of sparse and redundant representations in image processing*, Proceedings of the IEEE, 98 (2010), pp. 972–982.

[13] Ehsan Elhamifar and René Vidal, *Sparse subspace clustering*, in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 2790–2797.

[14] Ke Huang and Selin Aviyente, *Sparse representation for signal classification*, in Advances in neural information processing systems, 2006, pp. 609–616.

[15] Kenneth Kreutz-Delgado, Joseph F Murray, Bhaskar D Rao, Kjersti Engan, Te-Won Lee, and Terrence J Sejnowski, *Dictionary learning algorithms for sparse representation*, Neural computation, 15 (2003), pp. 349–396.

[16] David Asher Levin, Yuval Peres, and Elizabeth Lee Wilmer, *Markov chains and mixing times*, AMS Bookstore, 2009.

[17] Zhouchen Lin, Minming Chen, and Yi Ma, *The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices*, arXiv preprint arXiv:1009.5055, (2010).

[18] Zhouchen Lin, Risheng Liu, and Zhixun Su, *Linearized alternating direction method with adaptive penalty for low-rank representation*, in Advances in Neural Information Processing Systems, 2011, pp. 612–620.

[19] Haifeng Liu, Zheng Yang, and Zhaohui Wu, *Locality-constrained concept factorization*, in IJCAI Proceedings-International Joint Conference on Artificial Intelligence, vol. 22, 2011, p. 1378.

[20] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro, *Online dictionary learning for sparse coding*, in Proceedings of the 26th Annual International Conference on Machine Learning, ACM, 2009, pp. 689–696.

[21] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman, *Supervised dictionary learning*, in NIPS, 2008, pp. 1033–1040.

[22] Trygve Randen and John Hakon Husoy, *Filtering for texture classification: A comparative study*, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 21 (1999), pp. 291–310.

[23] Sam T Roweis and Lawrence K Saul, *Nonlinear dimensionality reduction by locally linear embedding*, Science, 290 (2000), pp. 2323–2326.

[24] Ron Rubinstein, Tomer Faktor, and Michael Elad, *K-svd dictionary-learning for the analysis sparse model*, in Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on, IEEE, 2012, pp. 5405–5408.

[25] Ron Rubinstein, Tomer Peleg, and Michael Elad, *Analysis k-svd: A dictionary-learning algorithm for the analysis sparse model*, Signal Processing, IEEE Transactions on, 61 (2013), pp. 661–677.

[26] Gehong Zhao, Guangmin Wu, Yue Liu, and Janming Chen, *Texture classification based on completed modeling of local binary pattern*, in Computational and Information Sciences (ICCIS), 2011 International Conference on, IEEE, 2011, pp. 268–271.